Remarks/Arguments

Office Action Summary

Status.

1. This *RESPONSE A* is in answer to the Office communication mailed 07/14/2005.
2. The Office communication is non-final.
3. NA

Disposition of Claims.

4. Claims 1 - 6 are pending in the application.
5. No Claims have been allowed.
6. The rejected Claims 1 - 6 have been amended.
7. NA
8. NA

Application Papers.

9. NA
10. The drawings have been amended.
11. NA

Priority under 35 U.S.C. § 119.

12. NA

DETAILED ACTION

1. Claims 1-6 as amended by this *RESPONSE A* are presented for reconsideration and examination. This *RESPONSE A* was mailed Post-Office-To-Addressee on Tuesday, January 17, 2006. January 14, January 15 and January 16 were Saturday, Sunday and Holiday (Martin Luther King Day), respectively.

1.1 **Drawings.** Alignment errors of text in the drawings have been corrected together with conforming the numbering to the specification.

1.2 **Specification.** The specification has been amended to conform the numbering to the drawings.

Claim Interpretation

2. The Examiner suggests an interpretation as follows:

2.1 Office action Page 2:

> *Claim 1: State information is used for determining a "program mode of operation". By applicant's own admission (Specification: Pg.6-7, [0027]) the features and modes are defined by the program execution information. Further, in S/390 architecture, program information is stored in Program status word (PSW) & control registers. Hence, examiner interprets that state information is stored in PSW & control registers as well. Further, state information is interpreted as a flag indicating if the code is ready for execution (i.e. is it already translated).*

2.2 To the extent that the Examiner's interpretation is understood, it is believed in error. As described in the Specification (Page 6-7 [0027]), the PSW and control register information changes from time to time during execution. The particular PSW information when legacy code is first translated to translated code may have first values, but when that legacy code again appears for execution a second time, the PSW information will have second values with no guarantee that the second values are the same as the first values. If different, execution of the already translated code based upon the PSW information second values will lead to incorrect execution since the translated code was based on PSW information first values. Accordingly, the required information for proper execution of the translated code is nowhere stored in the PSW and control registers.

2.3 Anything in the Examiner's interpretation quoted in Section 2.1 contrary to the above explanation in Section 2.2 is in error.

2.4 The Examiner's statement "*Hence, examiner interprets that state information is stored in PSW & control registers as well. Further, state information is interpreted as a flag indicating if the code is ready for execution (i.e. is it already translated)*" appearing as the end of Section 2.1, to the extent inconsistent with the explanation in Section 2.2 is in error. Further, no state information is stored in the PSW and control registers that indicates whether translated code is properly available for execution with the correct information in the PSW and control registers.

Claim Rejections -35 USC § 101

3. The Examiner rejected Claims 1-5 under 35 U.S.C. §101 because the claimed invention is allegedly directed to non-statutory subject matter.

3.1 The citations of the Examiner are noted together with the suggested amendments to overcome the rejection.

3.2 The independent Claim 1 has been amended to recited a "computer-implemented method" and "executing said translated instructions to emulate said legacy instructions" and therefore the rejection under 35 U.S.C. §101 is believed overcome.

Claim Rejections -35 USC § 102

4. The Examiner rejected Claims 1-2, 4& 6 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,516,295 issued to George A. Mann et al (Mann '295 hereafter).

4.1 In making the rejection, the Examiner states (numbering added):

Regarding Claim 1

4.1.1 Mann '295 teaches a method for dynamic emulation of a legacy instruction of a legacy program (Mann '295: Col.2 Lines 44-51).

4.1.2 Mann '295 teaches state information is store in form of a tag (Mann '295: Col.5 Table 1).

4.1.3 Further, Mann '295 teaches that the legacy instruction and the state information are accessed for each particular legacy instruction (Mann '295: Col. 6 Lines 11-19).

4.1.4 Further, Mann '295 teaches querying if the one or more legacy instructions stored as a result of translation for an execution mode (Mann '295: Col.5 Table 1; Col.5 Lines 54-67; Col.6 Lines 1-12, 21-28).

4.1.5 Mann '295 also teaches if the code is not translated (and used) then it is translate into one or more instructions and stored in blocks (including tag/state, offset/index execution status information) (Mann '295: Fig.3; Col.8 Lines 33-49) and then the block goes through dynamic emulation execution.

4.2 The arguments of the Examiner regarding Claim 1 are traversed for the following reasons.

4.2.1 By way of background, the term "*program execution mode*" has the meaning described in the specification, for example, in paragraphs [0010] and [0027] (using paragraph numbers of the application as filed and not as published), as follows:

> *[0010] Emulation frequently is used when a CISC architecture is emulated on a RISC architecture. The legacy code of the program from the CISC architecture is processed to obtain the translated code for the translated program in the RISC architecture. The program behavior in any computer architecture is not only a function of the code being executed, but also depends on various modes of operation that perform different functions which may be enabled or disabled at the time of execution. Mere inspection of the code itself does not reveal whether such modes of operation are being invoked. Therefore, the translation process is hampered and impacted by when only conventional instruction by instruction translation is employed.*
>
> *[0027] The CISC blocks 3_C-10 and 3_C-11 are understood to be executed as a function of various features and modes of operation available in the legacy system. These features and modes are defined by Program Execution Information. In the S/390 architecture, Program Execution Information is stored, for example, in the architecturally defined Program Status Word (PSW) and Control Registers. Mere inspection of the CISC code in the FIG. 3 translation example does not reveal all of the modes of execution that are possibly enabled.*

4.2.2 From the specification, and particularly the quotes in Section 4.2.1 above, "*program execution mode*" in the present application is referring to the overall architectural mode of the computer as it relates to programs being executed. These program execution modes are set, for example in the S/390 architecture in the Program Status Word (PSW), and are not decipherable by inspecting the individual instructions or any instruction state information used for individual instructions.

4.2.3 The Examiner relies upon the Table 1of Mann '295 and states *Mann '295 teaches state information is store in form of a tag (Mann '295: Col.5 Table 1).* The information in Table 1of Mann '295, is not "*state information for determining a program execution mode*". Rather,

the information in Table 1of Mann '295, such as state information in target code tag 72, is used in connection with a single instruction, not a *program execution mode.* The information in Table 1of Mann '295 is readily discernable by mere inspection (see target system code array 70 in FIG 3 of Mann '295). The problem addressed by the present invention is one where mere inspection of the code itself, unlike in Mann '295, does not reveal whether the program execution mode is being invoked. To the extent that the Examiner relies on the statement quoted in Section 4.1.2 above to teach applicant's Claim 1 element *"providing state information for determining a program execution mode* ", that reliance is traversed for the reasons explained.

4.2.4 Additionally, it is apparent from Mann '295 that there is no suggestion of storing the information in Table 1of Mann '295 into the Host Code translated store 88. There is no such suggestion because Mann '295 is concerned with a different problem. By way of contrast, Applicant's Claim 1 recites storing the state information with the translated instruction, to wit, "*storing said one or more particular translated instructions* ***with said state information***". This storing appears in applicant's FIG 4, for example, where "0000000000004200" is stored with the translated code. There is no corresponding storing in Mann '295.

4.2.5 In summary, Mann '295 is materially different from the present invention in that Mann '295 stores information only for the target (legacy) instructions and stores that information with the **target instructions** (not the translated instructions) and Mann '295 is not concerned with "*program execution mode*" information. By contrast, applicant's invention stores *program execution mode* information with the **translated instructions**. The result is that in applicant's invention, the same legacy instructions may have different corresponding translated instructions where the different ones of the translated instructions are stored with different *program execution mode* information. Such operation is neither suggested or possible with Mann '295.

4.3 The Examiners characterization of Mann '295 for Claim 2 is traversed as follows:

4.3.1 The Examiner states,

Regarding Claim 2

> *Mann '295 teaches the step of storing translation indications for only a subset of all translated blocks (Mann '295: Col.5 Lines 54-63). Further, Mann '295 teaches that the state information/tag is stored in particular translated blocks (Mann '295: Fig.3 Elements 70, 80, 85 & 88).*

4.3.2 Mann '295 does not teach *that the state information/tag is stored in particular translated blocks*. Referring to Mann '295: Fig.3 Elements 70, 80, 85 & 88 identified by the Examiner, all of the "*state information/tag*" is stored in the target code array 70 which is the target instruction store and not the translated instructions store (Host Code store 88). The tables 80 in FIG 3 of Mann '295 are extensions of array 70 for indirect addressing into Host Code store 88. The line 85 according to Mann '295 is a forward chain linking block addresses. Nowhere does Mann '295 suggest storing any of the state information from target code array 70 with the translated instructions in the store 88.

4.3.3 In summary, Claim 2 differs as does Claim 1 in that Mann '295 stores information only for the target (legacy) instructions and stores that information with the **target instructions** (not the translated instructions) and Mann '295 is not concerned with "*program execution mode*" information. By contrast, applicant's invention stores *program execution mode* information with the **translated instructions**.

4.4 The Examiners rejection of Claim 4 is traversed for the same reasons as Claims 1 and 2 above.

4.5 The Examiners rejection of Claim 6 is traversed for the same reasons as Claims 1 and 2 above.

Claim Rejections -35 USC § 103

5. The Examiner rejected Claims 3 & 5 under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,516,295 issued to George A. Mann et al (Mann '295 hereafter) in view of U.S. Patent No. 5,560,013 issued to Casper A. Scalzi et al (Scalzi '013 hereafter).

5.1 The Examiner's rejection of Claims 3 & 5 is traversed for the same reasons as Claims 1 and 2 above.

Respectfully submitted,

SIGNATURE OF ATTORNEY		
David E. Lovejoy (Reg. No. 22,748)	David E. Lovejoy David E. Lovejoy	**Signature Date:** 17 January 2006
Address	Customer No.	Communication
102 Reed Ranch Rd. Tiburon, CA 94920-2025	21603	Tel: (415) 435-8203 Fax: (415) 435-8857 e-mail: david.lovejoy@sbcglobal.net

Amendments to the Drawings

The attached sheets 1/3, 2/3 and 3/3 of drawings includes changes to FIG. 2, FIG. 3 and FIG. 4 and replace the original sheets 1/3, 2/3 and 3/3.

In FIG. 2, previously numbered element 10 has been renumbered 11 and conforms to the numbering in the specification. In FIG. 3 and FIG. 4, letters split across two lines have been amended to appear on one line. For example, "$^{AD}_{D}$" has been replaced with "ADD".

Attachments:

Replacement Sheets 1/3, 2/3 and 3/3

Annotated Replacement Sheets 1/3, 2/3 and 3/3

FIG. 1

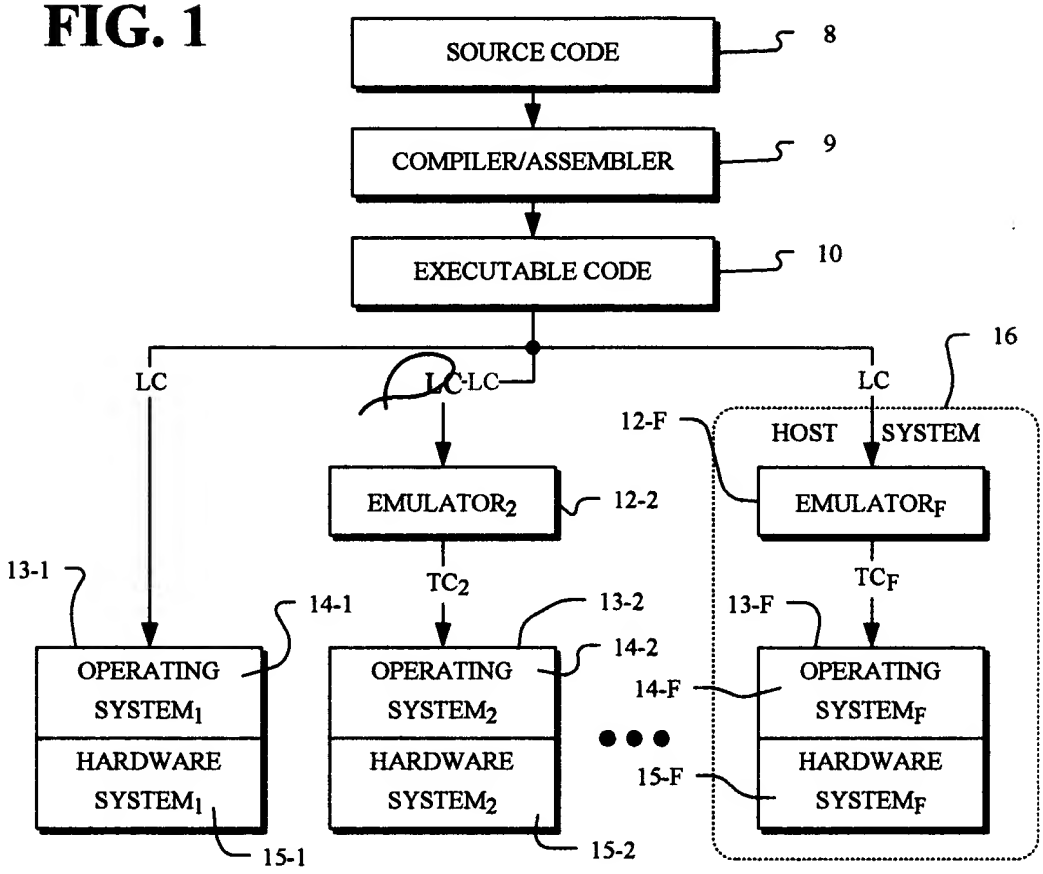
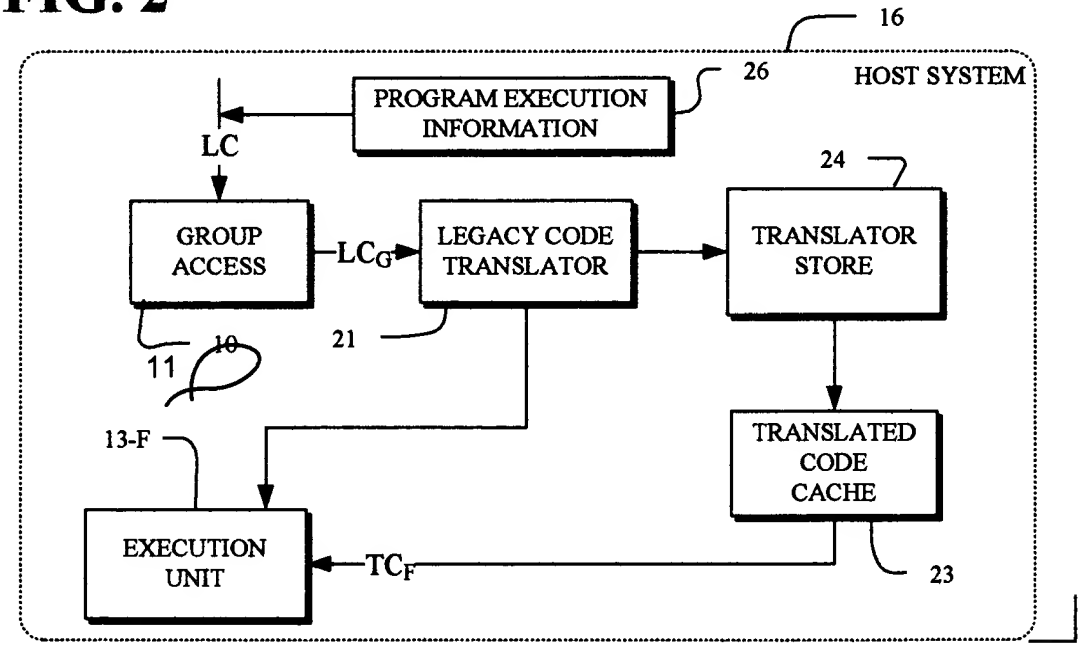


FIG. 2



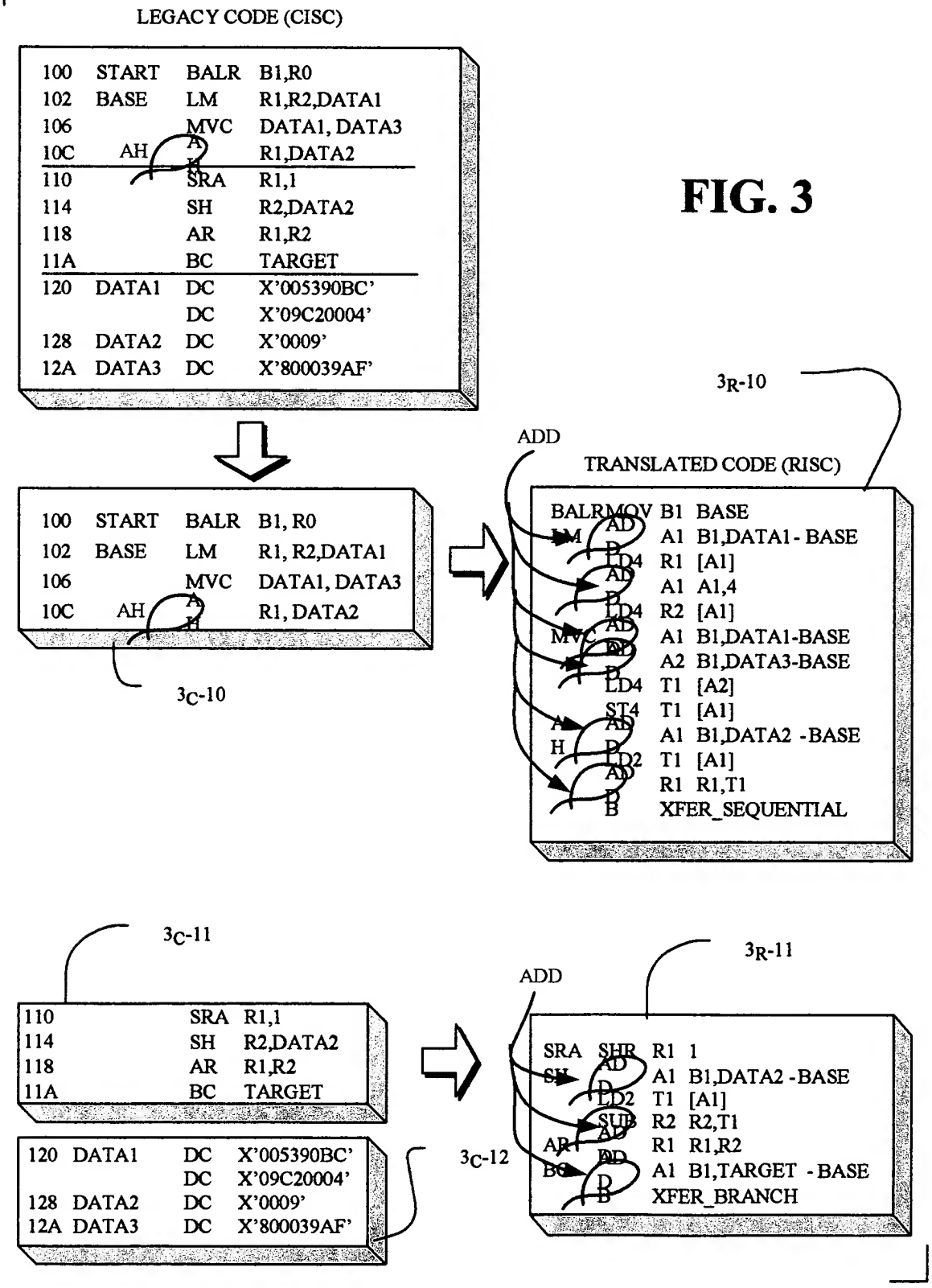
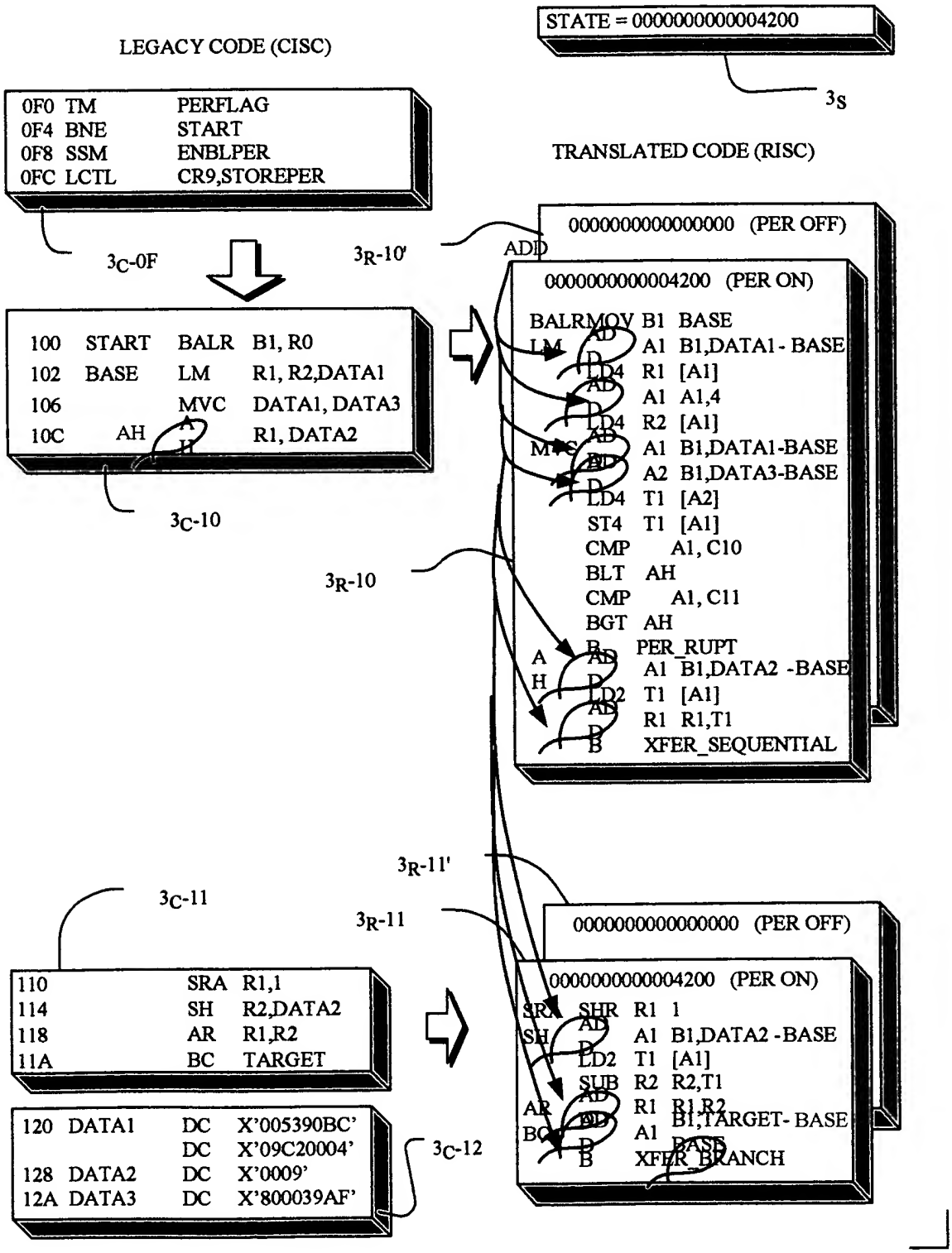

LEGACY CODE (CISC)
100 START BALR B1,R0
102 BASE LM R1,R2,DATA1
106 MVC DATA1, DATA3
10C AH R1,DATA2
110 SRA R1,1
114 SH R2,DATA2
118 AR R1,R2
11A BC TARGET
120 DATA1 DC X'005390BC'
DC X'09C20004'
128 DATA2 DC X'0009'
12A DATA3 DC X'800039AF'
100 START BALR B1, R0
102 BASE LM R1, R2,DATA1
106 MVC DATA1, DATA3
10C AH R1, DATA2
3C-10
3R-10
ADD
TRANSLATED CODE (RISC)
BALR MOV B1 BASE
LM AD A1 B1,DATA1 - BASE
D LD4 R1 [A1]
AD A1 A1,4
D LD4 R2 [A1]
MVC AD A1 B1,DATA1-BASE
D A2 B1,DATA3-BASE
D LD4 T1 [A2]
ST4 T1 [A1]
AH AD A1 B1,DATA2 -BASE
D LD2 T1 [A1]
AD R1 R1,T1
D B XFER_SEQUENTIAL
3C-11
110 SRA R1,1
114 SH R2,DATA2
118 AR R1,R2
11A BC TARGET
120 DATA1 DC X'005390BC'
DC X'09C20004'
128 DATA2 DC X'0009'
12A DATA3 DC X'800039AF'
3C-12
3R-11
ADD
SRA SHR R1 1
SH AD A1 B1,DATA2 -BASE
D LD2 T1 [A1]
SUB R2 R2,T1
AD R1 R1,R2
AR BC AD A1 B1,TARGET - BASE
D B XFER_BRANCH

FIG. 3

FIG. 4


STATE = 0000000000004200
3S
LEGACY CODE (CISC)
0F0 TM PERFLAG
0F4 BNE START
0F8 SSM ENBLPER
0FC LCTL CR9,STOREPER
3C-0F
100 START BALR B1, R0
102 BASE LM R1, R2,DATA1
106 MVC DATA1, DATA3
10C AH R1, DATA2
3C-10
TRANSLATED CODE (RISC)
0000000000000000 (PER OFF)
3R-10'
ADD
0000000000004200 (PER ON)
BALRMOV B1 BASE
LM AD A1 B1,DATA1 - BASE
D LD4 R1 [A1]
AD A1 A1,4
D LD4 R2 [A1]
MVC AD A1 B1,DATA1-BASE
AD A2 B1,DATA3-BASE
D LD4 T1 [A2]
ST4 T1 [A1]
CMP A1, C10
BLT AH
CMP A1, C11
BGT AH
B PER_RUPT
3R-10
A AD A1 B1,DATA2 -BASE
H D LD2 T1 [A1]
AD R1 R1,T1
D B XFER_SEQUENTIAL
3R-11'
3C-11
3R-11
0000000000000000 (PER OFF)
110 SRA R1,1
114 SH R2,DATA2
118 AR R1,R2
11A BC TARGET
0000000000004200 (PER ON)
SRA SHR R1 1
SH AD A1 B1,DATA2 -BASE
D LD2 T1 [A1]
SUB R2 R2,T1
AR AD R1 R1,R2
BC AD A1 B1,TARGET- BASE
D BASE
B XFER_BRANCH
120 DATA1 DC X'005390BC'
DC X'09C20004'
128 DATA2 DC X'0009'
12A DATA3 DC X'800039AF'
3C-12